

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | | |
|--|---|------------------------|
| In re application of | : | |
| | : | |
| MICHEL SHANE SIMPSON et al. | : | |
| | : | Examiner: Anh Ly |
| Serial No.: 09/943,786 | : | Group Art Unit: 2162 |
| | : | Confirmation No.: 1045 |
| Filed: August 31, 2001 | : | |
| | : | |
| For: METHOD AND APPARATUS FOR PRESENTING, SEARCHING, AND VIEWING DIRECTORIES | : | |

AFFIDAVIT UNDER 37 CFR § 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

We, Michel Shane Simpson, Brett Dee Garrett, Nathan Blaine Jensen, and William Donald Peterson, III, do hereby declare and say:

1. That we are named co-inventors in the above-identified patent application.
2. That we are familiar with the prosecution of this application, and understand that claims 21-30 and 32-40 are pending, claims 21-25, 26-29, and 31-41 (of which claims 31 and 41 have been canceled) were rejected as being obviously unpatentable over U.S. Patent No. 5,758,343 to Vigil (hereinafter "Vigil") in view of U.S. Patent Publication No. 2002/0032775 to Venkataramaiah et al. (hereinafter "Venkataramaiah"), and claims 26 and 30 were rejected as obvious in view of the Vigil and Venkataramaiah combination, in further

view of U.S. Patent Publication No. 2006/0129652 to Petrovskaya.

3. That Venkataramaiah was filed on August 28, 2001 and published on March 14, 2002, and that the Venkataramaiah reference relates to “transmitting and retrieving data via a distributed persistence framework.” *Venkataramaiah Title*.

4. That prior to August 28, 2001, the effective filing date of Venkataramaiah, the invention described and claimed in the pending application was completed by us and/or under our direction in the United States.

5. That prior to August 28, 2001, which is three days before the filing date of the pending application, we completed or had completed under our direction: a) on a computer, a directory shell with an administrator utility and a directory browser, b) enabling the administrator utility to associate, such as with a table, directory classes of two or more disparate directories into a single user-searchable category, such as by way of mapping a category attribute of the single user-searchable category to one or more class attributes of the directory class, c) from the directory browser, enabling direct searching of the directory classes of the two or more disparate directories with a single query of the single user-searchable category, and d) displaying search results of the single query, such as on a panel of the directory browser; which establishes conception and reduction to practice of the invention claimed in this application and that such conception and reduction to practice is evidenced by Exhibits A, B, and C, wherein they include:

A. A two page invention disclosure (labeled IDR-530), with various dates and name information redacted, submitted to the Novell, Inc. (Assignee of the entire right, title and interest) invention disclosure committee, whereby the title and attendant written description indicates, among other things in the paragraph numbered 4:

1) a "DirectoryScheme architecture ... to facilitate permanent attribute/class groupings and subsequent lookup" and "DirectoryScheme objects" as representatively found in claims 21-30 and 32-40;

2) "treating [the] objects as one and rendering the designated attributes together in one convenient search" as representatively found in claims 21, 25, 28, 29, 32, and 37;

3) Including, the "administrator now [having] the ability to group objects together in an ObjectScheme to meet his exact needs," as representatively found in claims 21-30 and 32-40;

4) Each "DirectoryScheme object (one per defined directory) will house Admin-defined ObjectScheme tables, which, in turn, will house all corresponding AttributeScheme objects" with an "ObjectScheme" consisting of "Schema class names, an ObjectScheme keyname and a superset of all class attributes contained therein ..., Ie [sic] A "USER" object scheme for NDS could include the following classes, but not be limited to: InetOrgPerson and Person;"

5) An "attribute table lookup for the client ... [in] a singleton class;"

- 6) "API's" as representatively found in claims 23, 26, 30, and 32;
- 7) "Directory attribute settings" and "AttributeScheme objects" as representatively found in claims 25 and 29; and
- 8) "Admin-defined ObjectScheme tables" as representatively found in claims 27 and 32;

B. A one page email, with various dates and name information redacted, submitting the invention disclosure of Exhibit A to the Novell, Inc. invention disclosure committee, and including the description of Exhibit A; and

C. A two page email, with various dates and name information redacted, acknowledging the receipt of the invention disclosure of Exhibit A by the Novell, Inc. invention disclosure committee.

6. That Exhibits A, B, and C individually and/or collectively indicate conception and reduction to practice of the present invention prior to the effective date of Venkataramaiah.

7. The undersigned further declares that all statements made herein of his own knowledge are true and that all statements made on information and belief are believed to be true; and further than these statements were made with the knowledge that willful false statements and the like so made and punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may

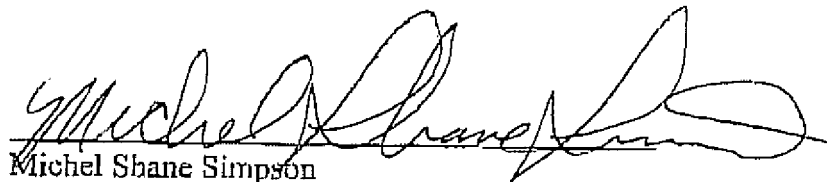
Appl. Ser. No. 09/943,786

Affidavit Under 37 CFR 1.131

Reply to Final Office Action dated January 11, 2008

jeopardize the validity of the application or any patent issuing therefrom.

3/28/08
Date


Michel Shane Simpson

Date

Brett Dee Garrett

Date

Nathan Blaine Jensen

Date

William Donald Peterson, III

Appl. Ser. No. 09/943,786
Affidavit Under 37 CFR 1.131
Reply to Final Office Action dated January 11, 2008

jeopardize the validity of the application or any patent issuing therefrom.

Date

1 April 2008

Date

Date

Date

Michel Shane Simpson

Brett Dee Garrett

Brett Dee Garrett

Nathan Blaine Jensen

William Donald Peterson, III

Appl. Ser. No. 09/943,786
Affidavit Under 37 CFR 1.131
Reply to Final Office Action dated January 11, 2008

jeopardize the validity of the application or any patent issuing therefrom.

Date

Michel Shane Simpson

Date

Brett Dee Garrett

April 4, 2008

Date

Nathan Blaine Jensen

Nathan Blaine Jensen

Date

William Donald Peterson, III

Appl. Ser. No. 09/943,786
Affidavit Under 37 CFR 1.131
Reply to Final Office Action dated January 11, 2008

jeopardize the validity of the application or any patent issuing therefrom.

Date

Michel Shane Simpson

Date

Brett Dee Garrett

Date

Nathan Blaine Jensen

April 1, 2008
Date

William Donald Peterson, III
William Donald Peterson, III

EXHIBIT A

NOVELL PATENT PROGRAM INVENTION DISCLOSURE

E-mail your completed form and any attachments to PATENT PROGRAM. You'll hear from us within a couple of days. If you don't, something went wrong and you should call [REDACTED] to straighten it out.

Novell's Inventions Committee looks forward to reviewing ALL submitted invention disclosures and plans to review yours at our meeting in the middle of next month. If you plan to publicly disclose the invention before then, please call [REDACTED] and she'll move it up in the queue. Thank you for helping to protect Novell's intellectual property!

For more information on how the Patent Program works and to pick out the Novell Patent shirt that you want, see our website on the Innerweb!

1. What's the name of your invention?

DIRECTORYSCHEME TECHNOLOGY

2. Who has been involved in creating the invention and what did they contribute? What group are they in and who is their manager [REDACTED]

Mike Simpson: Designer, author

Brett Garrett: Feedback, customer needs review

Nate Jensen: Code/Design review/Admin code implementation

Will Peterson: Dynamic web content, XSL implementation

We all work for the eGuide team.

James Whitchurch is the director over this team.

3. Does Novell plan to use this invention in a product, pursue the invention in a standards body, or publish an article about it? Identify the product, standards organization, or publication. If you don't know, just say what you think Novell should do with it.

Net Directory. We are using this technology with eGuide to group LDAP classes/attribute information. The concept is to treat the grouping of multiple objects and corresponding attributes as one.

4. What is the purpose of your invention? What problem were you trying to solve when you developed this invention? Specify the pieces of your invention that are original and new (thus, patentable). How does your invention work? Diagrams and flowcharts go along way toward helping us quickly understand your invention, so be sure to include some.

DirectoryScheme Architecture Overview

Since eGuide has a requirement to support multiple directories on the back-end simultaneously, the DirectoryScheme architecture has been developed to facilitate permanent attribute/class groupings and subsequent lookup.

All Directory attribute settings (Admin-defined and Schema Information) will ultimately reside in the top-level DirectoryScheme object. Each DirectoryScheme object (one per defined directory) will house Admin-defined ObjectScheme tables, which, in turn, will house all corresponding AttributeScheme objects.

An ObjectScheme will consist of a collection of Schema class names, an ObjectScheme keyname and a superset of all class attributes contained therein. I.e. A "USER" object scheme for NDS could include the following classes, but not be limited to: InetOrgPerson and Person. Through the DirectoryScheme API's the Administrator can dynamically add or remove any additional ObjectScheme classes to a DirectoryScheme, which, in turn, would dynamically create/update its internal table of all related AttributeScheme objects, which can later be accessed for reading/modification. Also, the Administrator, through ObjectScheme API's, can dynamically add or remove class definitions to an existing ObjectScheme or can create new ObjectSchemes to meet his own defined needs. Class manipulation of

DOCKETED

DATE [REDACTED]

ObjectSchemes will also dynamically update the internal table of AttributeSchemes to reflect new changes.

This DirectoryScheme architecture, for example, provides the ability to perform a search on a person and a printer simultaneously, in effect, treating these objects as one and rendering the designated attributes together in one convenient search. The administrator now has the ability to group any objects together in an ObjectScheme to meet his exact needs.

Ultimately, attribute table lookup for the client will occur with a singleton class, Schemer. ObjectScheme objects will be referenced by KeyName, which is required during creation. AttributeScheme access can be accomplished through KeyName or Schema attribute name. Since eGuide requires a KeyName for each attribute due to supporting multiple Directories with varying syntax, a client request coming in with the corresponding DirectoryID and ObjectSchemeID can easily be serviced through the following syntax:

`Schemer.getInstance().getDirectoryScheme(Directory.uid).getObjectScheme(Object.uid).getAttributeByKey("LASTNAME").` Once this AttributeScheme object has been retrieved, any contained Schema/Admin Settings attribute criteria can easily be extracted/modified through the AttributeScheme API's.

5. If you are aware of others in the industry who have attempted to solve this problem before, describe why your solution is better than theirs (don't be modest!). eGuide 1.5 does not support this functionality. We are not aware of any other product that provides this "ObjectScheme" functionality.

EXHIBIT B

From: [REDACTED]
To: Brett Garrett; Mike Simpson; Nate Jensen; Will Peterson
Date: [REDACTED]
Subject: IDR-530: DIRECTORY SCHEME TECHNOLOGY

Privileged and Confidential

RE: IDR-530: DIRECTORY SCHEME TECHNOLOGY

Hi Mike, Brett, Will, and Nate;

Thank you for submitting your invention disclosure. It has been assigned the reference number IDR-530: DIRECTORY SCHEME TECHNOLOGY.

The Inventions Committee should review your IDR at their meeting in [REDACTED] unless necessary to do so sooner, and you will be contacted once a decision has been reached. (approximately the first week in [REDACTED]).

Each IDR submitted entitles you to [REDACTED]

We have 6 different shirts to choose from: a white polo (in XXL only), Steel blue polo (no picture), a teal polo, a maroon polo, a long sleeved sage rugby (in XL only), and a long sleeved denim shirt; or we also have navy blue hats. You can view the shirts on our web site at:

http://innerweb.novell.com/groups/legal/Patent_Site/Patent_Recognition/patent_recognition.html

If you have any questions, please let me know.

Thank you!!

[REDACTED]
Legal-Patent Assistant

[REDACTED]
Novell, Inc., the leading provider of Net services software.
www.novell.com

This e-mail message may contain confidential and privileged material for the sole use of the intended recipients. Review, dissemination, or other use by anyone else is strictly prohibited. If you are not an intended recipient, please contact the sender and delete all copies.

CC: [REDACTED]

DOCKETED

DATE [REDACTED]

EXHIBIT C

From: <nobody@novell.com>
To: [REDACTED]
Date: [REDACTED]
Subject: IDRform

[REDACTED]
NAME= Mike Simpson
EMAIL= misimpson@novell.com
INVENTIONNAME= DirectoryScheme technology
CONTRIBUTE_MANAGER= Mike Simpson: Designer, author
Brett Garrett: Feedback, customer needs review
Nate Jensen: Code/Design review/Admin code implementation
Will Peterson: Dynamic web content, XSL implementation

We all work for the eGuide team.

James Whitchurch is the director over this team.

NOVELLSPLANS= Net Directory. We are using this technology with eGuide to group LDAP classes/attribute information. The concept is to treat the grouping of multiple objects and corresponding attributes as one.

PURPOSE= DirectoryScheme Architecture Overview

Since eGuide has a requirement to support multiple directories on the back-end simultaneously, the DirectoryScheme architecture has been developed to facilitate permanent attribute/class groupings and subsequent lookup.

All Directory attribute settings (Admin-defined and Schema information) will ultimately reside in the top-level DirectoryScheme object. Each DirectoryScheme object (one per defined directory) will house Admin-defined ObjectScheme tables, which, in turn, will house all corresponding AttributeScheme objects. An ObjectScheme will consist of a collection of Schema class names, an ObjectScheme keyname and a superset of all class attributes contained therein. I.e. A "USER" object scheme for NDS could include the following classes, but not be limited to: InetOrgPerson and Person. Through the DirectoryScheme API's the Administrator can dynamically add or remove any additional ObjectScheme classes to a DirectoryScheme, which, in turn, would dynamically create/update its internal table of all related AttributeScheme objects, which can later be accessed for reading/modification. Also, the Administrator, through ObjectScheme API's, can dynamically add or remove class definitions to an existing ObjectScheme or can create new ObjectSchemes to meet his own defined needs. Class manipulation of ObjectSchemes will also dynamically update the internal table of AttributeSchemes to reflect new changes.

This DirectoryScheme architecture, for example, provides the ability to perform a search on a person and a printer simultaneously, in effect, treating these objects as one and rendering the designated attributes together in one convenient search. The administrator now has the ability to group any objects together in an ObjectScheme to meet his exact needs.

Ultimately, attribute table lookup for the client will occur with a singleton class: Schemer. ObjectScheme objects will be referenced by KeyName, which is required during creation. AttributeScheme access can be accomplished through KeyName or Schema attribute name. Since eGuide requires a KeyName for each attribute due to supporting multiple Directories with varying syntax, a client request coming in with the corresponding DirectoryID and ObjectSchemeID can easily be serviced through the following syntax: Schemer.getInstance().getDirectoryScheme(Directory.uid).getObjectScheme(Object.uid).getAttributeByKey("LASTNAME"). Once this AttributeScheme object has been retrieved, any contained Schema/Admin Settings attribute criteria can easily be extracted/modified through the AttributeScheme API's.

BETTERSOLUTION= eGuide 1.5 does not support this functionality. We are not aware of any other product that provides this "ObjectScheme" functionality.

FormsButton1= Submit

2008-08-08 10:00 AM